DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office Action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 4, 2011 has been entered.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on April 4, 2011 was considered by the examiner.

Response to Amendment

1. Claims 15-17, 20-22 and 24-27 are pending.

2. The objection to the Specification is withdrawn. The objection to Figure 6 is maintained, because the Specification at page 2 line 5-7 discloses that the figure relates to the prior art.

Drawings

3. Figure 6 should be designated by a legend such as --Prior Art-- because of the disclosure within the Specification at page 2 and because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR

1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled “Replacement Sheet” in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

4. Claims 15, 21 and 24 are objected to because of the following informalities: the claims recite "are capable of being independently from the other ..." However, independently does not refer to anything. It appears that from the previously presented claims that the claims should read "are capable of being executed independently from the other…" Appropriate action is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

7. Claims 15-17, 20-22 and 24-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Han et al. ("Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, Pages 53-87, hereinafter referred to as Han) in view of Agrawal et al. (U.S. Patent No. 6,230,151 B1, hereinafter referred to as "Agrawal") and in further view of Zaki et al., "Parallel Classification for Data Mining on Shared-Memory Multiprocessors," pages 1-8, hereinafter referred to as "Zaki").

As per claim 15, Han discloses **a method for mining data of a database, comprising:**

identifying transaction items of the database and determining an occurrence frequency for each item; (a scan of DB derives a list of frequent items, <(f:4), (c:4), (a:3), (B:3), (m:3), (p:3)>. The frequent items are listed in this ordering in the rightmost column of Table 1.)(e.g. page 57; see also, pages 56 and 58);

Han does not specifically disclose **locking the identified transaction items to prevent other data mining processes from selecting the identified transaction items;**

Han discloses **building a probe structure based on the identified frequent transaction items;** (e.g. algorithm 1 – sort frequent items in support-descending order and generates figure 1)(page 58 and page 60)

Han discloses **building a plurality of disjoint branches for the probe structure,** (tree may consist of a single prefix path of FP-tree, tree may also consist of a multipath portion and frequent patterns can be partitioned into three portions: the single prefix-path portion P, the multipath portion Q and their combinations (e.g. page 65, Figure 4 and page 68)

Han discloses **wherein each branch of the probe structure includes a number of identified transaction items selected based on content of the transaction items and the occurrence frequency of the transaction items,** (e.g. algorithm 1 – sort frequent items in support-descending order and generates figure 1)(page 58 and page 60)

Although Han discloses that a single path and since every item in each path is unique and there is no redundant pattern to be generated (page 66), Han does not specifically disclose **at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of being** *(executed)* **independently from the other plurality of disjoint branches;**

Han discloses **grouping the branches of the probe structure based on the content of the transaction items of each branch;** (the B+-tree structure, can be used to index FP-tree as well. Since there are many operations localized to single paths or individual item prefix sub-trees, such a pattern matching for node insertion, creation of transformed prefix paths for each node, it is important to cluster FP-tree nodes according to the tree/subtree structure.)(page 81)

Han discloses **building a frequent pattern tree (FP-tree) from the branches of the probe structure;** (examples of fp trees are shown in Figs. 1-4)(pages 58, 60, 62 and 65)

Han does not specifically disclose **assigning, via a master processor, each branch of the FP-tree to one of a plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.**

On the other hand, Agrawal, which discloses a parallel classification for data mining in a shared-memory multiprocessor system, does disclose **locking the identified transaction items to prevent other data mining processes from selecting the identified transaction items;** (examples of locking include dynamic scheduling of assigning d attribute lists to P processors with good load balance and small overhead can be applied. Dynamic subtree task parallelism and mutually exclusive access via locking)(e.g. col 10 lines 18-21 and col 13 lines 19-20 and 35-37)

at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of being *(executed)* **independently from the other plurality of disjoint branches;** (multiple processor groups work on distinct subtrees (branches). If only one leaf remains, all processors are assigned to that leaf. If multiple leaves and multiple processors, the group master splits the processor set into two parts, and also splits the leaves into two parts.)(e.g. col 13 lines 43-51)

On the other hand, Zaki, which discloses shared memory parallelization of decision tree construction, discloses **assigning, via a master processor, each branch of the FP-tree to one of a plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.** (the task parallel approach uses dynamic sub-tree partitioning among processors. A master is responsible for partitioning the processor set. Each group independently executes processes on distinct sub-trees. Substantially equal is considered to be satisfied by task pipelining and dynamic load balancing to yield faster implementations)(Abstract, pages 1, 3 and 5).

Han relates to data mining using frequent pattern trees. Abstract. On the other hand, Agrawal discloses parallel classification for data mining in a shared-memory multiprocessor system. E.g. abstract. The parallel classification provides for splitting the processes and allowing for increased processing speed. E.g. col 3 lines 53-56. Zaki teaches data mining and data parallelism in decision trees. E.g. pages 1, 3 and 5 The parallelism disclosed in Zaki increase the efficiency of data-mining. Abstract on page 1. Zaki discloses using a master and slave and dynamic load balancing to yield faster implementations. Abstract, pages 1, 3 and 5

It would have been obvious to one of the ordinary skill in the art at the time of the applicant's invention to incorporate the use of the master and slave processors of Zaki to the device of Han because both relate to data-mining and decision trees [Han:

Abstract; Agrawal: abstract and figure 8; Zaki: Abstract, 1, 3 and 5]. Because Han, Agrawal and Zaki relate to decision trees and data mining, it would have been obvious to one of the ordinary skill in the art at the time of the applicant's invention to substitute and/or modify one method for the other to achieve the predictable result of improving dynamic load balancing for faster data mining functions.

As per claim 16, Han in view of Agrawal and in further view of Zaki disclose the method of claim 15. Han further discloses **further comprising scanning a first portion of the database when identifying transaction items of the database, and scanning a second portion of the database when building the probe structure, wherein the probe structure includes an associated number of counts with each branch of the probe structure after scanning the second portion of the database.** (count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name.)(page 57-58 and figure 1)

As per claim 17, Han in view of Agrawal and in further view of Zaki disclose the method of claim 15. Han further discloses **further comprising building the probe structure to include a probe tree and probe table, and using the probe tree and probe table to build the FP-tree for mining the FP-tree to determine frequent data patterns.** (the logic identifies the frequent items and segments each transaction into items and arranges the frequent items in descending order as shown in table I and

Example 1)(page 56-57)**,** (if two transactions share a common prefix, the shared parts can be merged using one prefix structure as long as the count is registered properly)(page 57)**.**

As per claim 20, Han in view of Agrawal and in further view of Zaki disclose the method of claim 15. Han further discloses **further comprising partitioning the database according to content of the identified transaction items to obtain the probe structure, wherein the probe structure includes combinations of the identified transaction items and the number of occurrences of one or more content-based transactions.** (the logic identifies the frequent items and segments each transaction into items and arranges the frequent items in descending order as shown in table I and Example 1)(page 56-57)**,** (if two transactions share a common prefix, the shared parts can be merged using one prefix structure as long as the count is registered properly)(page 57)**.**

As per claim 21, Han discloses **a computer-readable non-transitory storage medium having stored thereon instructions, which when executed in a system operate to manage data of a database by:** (page 55)

identifying transaction items of the database and determining an occurrence frequency for each item; (a scan of DB derives a list of frequent items, <(f:4), (c:4), (a:3), (B:3), (m:3), (p:3)>. The frequent items are listed in this ordering in the rightmost column of Table 1.)(e.g. page 57; see also, pages 56 and 58)**;**

Han does not specifically disclose **locking the identified transaction items to prevent other data mining processes from selecting the identified transaction items;**

Han discloses **building a probe structure based on the identified frequent transaction items;** (e.g. algorithm 1 – sort frequent items in support-descending order and generates figure 1)(page 58 and page 60)

Han discloses **building a plurality of disjoint branches for the probe structure,** (tree may consist of a single prefix path of FP-tree, tree may also consist of a multipath portion and frequent patterns can be partitioned into three portions: the single prefix-path portion P, the multipath portion Q and their combinations (e.g. page 65, Figure 4 and page 68)

Han discloses **wherein each branch of the probe structure includes a number of identified transaction items selected based on content of the transaction items and the occurrence frequency of the transaction items,** (e.g. algorithm 1 – sort frequent items in support-descending order and generates figure 1)(page 58 and page 60)

Although Han discloses that a single path and since every item in each path is unique and there is no redundant pattern to be generated (page 66), Han does not specifically disclose **at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of being** *(executed)* **independently from the other plurality of disjoint branches;**

Han discloses **grouping the branches of the probe structure based on the content of the transaction items of each branch;** (the B+-tree structure, can be used to index FP-tree as well. Since there are many operations localized to single paths or individual item prefix sub-trees, such a pattern matching for node insertion, creation of transformed prefix paths for each node, it is important to cluster FP-tree nodes according to the tree/subtree structure.)(page 81)

Han discloses **building a frequent pattern tree (FP-tree) from the branches of the probe structure;** (examples of fp trees are shown in Figs. 1-4)(pages 58, 60, 62 and 65)

Han does not specifically disclose **assigning, via a master processor, each branch of the FP-tree to one of a plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.**

On the other hand, Agrawal, which discloses a parallel classification for data mining in a shared-memory multiprocessor system, does disclose **locking the identified transaction items to prevent other data mining processes from selecting the identified transaction items;** (examples of locking include dynamic scheduling of assigning d attribute lists to P processors with good load balance and small overhead can be applied. Dynamic subtree task parallelism and mutually exclusive access via locking)(e.g. col 10 lines 18-21 and col 13 lines 19-20 and 35-37)

at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of being *(executed)* **independently from the other plurality of disjoint branches;** (multiple processor groups work on distinct subtrees (branches). If only one leaf remains, all processors are assigned to that leaf. If multiple leaves and multiple processors, the group master splits the processor set into two parts, and also splits the leaves into two parts.)(e.g. col 13 lines 43-51)

On the other hand, Zaki, which discloses shared memory parallelization of decision tree construction, discloses **assigning, via a master processor, each branch of the FP-tree to one of a plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.** (the task parallel approach uses dynamic sub-tree partitioning among processors. A master is responsible for partitioning the processor set. Each group independently executes processes on distinct sub-trees. Substantially equal is considered to be satisfied by task pipelining and dynamic load balancing to yield faster implementations)(Abstract, pages 1, 3 and 5).

It would have been obvious to combine Han with Agrawal and Zaki for the reasons stated in claim 15 above.

As per claim 22, Han and Zaki disclose the computer-readable non-transitory storage medium of claim 21. Han further discloses **wherein the instructions, which**

when executed in a system operate to manage data of a database further by building the probe structure to include a probe tree and probe table, and using the probe tree and probe table to build the FP-tree for mining the FP-tree to determine frequent data patterns. (the logic identifies the frequent items and segments each transaction into items and arranges the frequent items in descending order as shown in table I and Example 1)(page 56-57)**,** (if two transactions share a common prefix, the shared parts can be merged using one prefix structure as long as the count is registered properly)(page 57)**.**

As per claim 24, Han discloses **a system comprising:**

Zaki discloses **a master processor;** (e.g. page 5)

Zaki discloses **a plurality of slave processors;** (e.g. page 5)

Han discloses **a database; and** (abstract)

Han discloses **software to Identify transaction items of the database and determining an occurrence frequency for each item,** (a scan of DB derives a list of frequent items, <(f:4), (c:4), (a:3), (B:3), (m:3), (p:3)>. The frequent items are listed in this ordering in the rightmost column of Table 1.)(e.g. page 57; see also, pages 56 and 58)

Han does not specifically disclose **lock the identified transaction items to prevent other data mining processes from selecting the identified transaction items,**

Han discloses **build a probe structure based on the identified frequent transaction items;** (e.g. algorithm 1 – sort frequent items in support-descending order and generates figure 1)(page 58 and page 60)

build a plurality of disjoint branches for the probe structure, (tree may consist of a single prefix path of FP-tree, tree may also consist of a multipath portion and frequent patterns can be partitioned into three portions: the single prefix-path portion P, the multipath portion Q and their combinations (e.g. page 65, Figure 4 and page 68)

wherein each branch of the probe structure includes a number of identified transaction items selected based on content of the transaction items and the occurrence frequency of the transaction items, (tree may consist of a single prefix path of FP-tree, tree may also consist of a multipath portion and frequent patterns can be partitioned into three portions: the single prefix-path portion P, the multipath portion Q and their combinations (e.g. page 65, Figure 4 and page 68)

Although Han discloses that a single path and since every item in each path is unique and there is no redundant pattern to be generated (page 66), Han does not specifically disclose **at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of** *(executed)* **independently from the other plurality of disjoint branches,**

Han does disclose **group the branches of the probe structure based on the content of the transaction items of each branch,** (the B+-tree structure, can be used to index FP-tree as well. Since there are many operations localized to single paths or

individual item prefix sub-trees, such a pattern matching for node insertion, creation of transformed prefix paths for each node, it is important to cluster FP-tree nodes according to the tree/subtree structure.)(page 81)

Han does disclose **build a frequent pattern tree (FP-tree) from the branches of the probe structure, and** (examples of fp trees are shown in Figs. 1-4)(pages 58, 60, 62 and 65)

Han does not specifically disclose **assign, via the master processor, each branch of the FP-tree to one of the plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.**

On the other hand, Agrawal does disclose **lock the identified transaction items to prevent other data mining processes from selecting the identified transaction items,** (examples of locking include dynamic scheduling of assigning d attribute lists to P processors with good load balance and small overhead can be applied. Dynamic subtree task parallelism and mutually exclusive access via locking)(e.g. col 10 lines 18-21 and col 13 lines 19-20 and 35-37)

at least two branches include a common transaction item, and each of the plurality of disjoint branches are capable of *(executed)* **independently from the other plurality of disjoint branches,** (multiple processor groups work on distinct subtrees (branches). If only one leaf remains, all processors are assigned to that leaf.

If multiple leaves and multiple processors, the group master splits the processor set into two parts, and also splits the leaves into two parts.)(e.g. col 13 lines 43-51).

On the other hand, Zaki, which discloses shared memory parallelization of decision tree construction, discloses **assign, via the master processor, each branch of the FP-tree to one of the plurality of slave processors, the plurality of slave processors to execute the transaction items identified by the respective branch in parallel with each other, wherein the number of transaction items to be executed by each of the plurality of slave processors is substantially equal.** (the task parallel approach uses dynamic sub-tree partitioning among processors. A master is responsible for partitioning the processor set. Each group independently executes processes on distinct sub-trees. Substantially equal is considered to be satisfied by task pipelining and dynamic load balancing to yield faster implementations)(Abstract, pages 1, 3 and 5).

It would have been obvious to combine Han with Agrawal and Zaki for the reasons stated in claim 15 above.

Claims 25-27 are substantially similar to claims 16, 17 and 20 respectively; therefore, they are rejected under the same subject matter.

Response to Arguments

8. Applicant's arguments with respect to claims 15-17, 20-22 and 24-27 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

The prior art made of record, listed on form PTO-892, and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RICHARD BOWEN whose telephone number is (571)270-5982. The examiner can normally be reached on Monday through Friday 7:30AM - 4:00PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Kim can be reached on (571)272-7421. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/R. B./
Examiner, Art Unit 2157
May 17, 2011

/CHARLES KIM/
Supervisory Patent Examiner, Art Unit 2157

/R. T./

Primary Examiner, Art Unit 2167